RON

MAIL FROM BBN-TENEXA RCVD AT 4-AUG-75 0625-PDT
DATE: 4 AUG 1975 0919-EDT
FROM: BURCHFIEL AT BBN-TENEXA
SUBJECT: PACKET RADIO TEMPORARY NOTE #145
TO: ABRAMSON AT ISI, CERF AT ISI, DICKSON AT BBN, ECOM-F4 AT ISI,
TO: KAHN AT ISI, KLEINROCK AT ISI, KUO AT ISI,
TO: MAGNAVOX AT ISI, NAC AT ISI, NIELSON AT SRI-AI,
TO: PRSETD AT OFFICE-1, TOBAGI AT ISI
CC: BURCHFIEL, BEELER

FIVE PAGES FOLLOW, PRTN #145, ENTITLED:

CROSS-RADIO DEBUGGING OF PACKET RADIO UNITS

Cross-Radio Debugging of Packet Radio Units

PRTN 145

Michael Beeler, BBN

July 31, 1975

This paper discusses the debugging of Packet Radio Units (PRUs) over a radio link. The discussion is in two sections. Section I is general and assumes the PRU contains permanently resident code to perform debugging functions. This is the mode in which BBN's cross-network debugger for PDP-11s operates. Section II describes the implementation planned by Collins Radio Corporation, and comments on the impact of this implementation on some of the issues brought up in section I.

I Preliminary PRU debugging specs

We propose the following operation for messages to debug Packet Radio Units remotely (over the air). The following assumes that headers such as HTP have been specified elsewhere; the packet has now been delivered to the debugging routines in the PRU (or station). Initially there is no need to burden the PRU with complex debugging commands; examine and deposit are sufficient. Perhaps this situation will continue, either because experience proves that more complex functions can be adequately performed by several examines/deposits (possibly including overlays), or because experience may show that the PRU memory would be taxed to include permanently resident code for complex debugging functions.

Since the PRU is in a state of flux, the examine/deposit actions operate on a changing entity. Under such conditions, it is often important to make changes in a coordinated manner. As one example, two words in diffepent places of the HTP code might need changing, but the code might not be executable between the moment that one were changed and the other were changed. Similarly, two variables might be monitored, but unless they are read (examined) at the same moment, their values might be meaningless. Therefore, a multiple examine command is highly desirable, as is a multiple deposit command. Each of these should have the capability of affecting a moderately large block of words, or several words scattered through memory, or a combination of the two.

The following suggestion is intended to accomplish the multiple examine/deposit function. Several of the particulars have intentionally been left loosely specified. This is because a minor change in format could seriously affect the ease of implementation and the size of code in the PRU, which Collins is best able to judge; whereas any format is easily accomodated in the station.

The examine/deposit packet should have several "chunks." Each chunk has three fields: address to start at, count, and data block. A single word, probably the first word after other headers (HTP, etc.) says whether these chunks are to be examined or deposited. Conceivably chunks could have a fourth field specifying examine/deposit on a per chunk basis, but we suspect this is unnecessary. We would appreciate comment from others on this point.

We would also appreciate advice from Collins on what the minimally addressible unit should be. Hopefully it can be a 16-bit word, which means only 12 bits are needed for the whole of the PRU's 4096-word memory. (12 bits permits a convenient packing of address and count fields, described below. Also, whole-word-only operation considerably

SIMPLIFIES THE STATION END OF THE DEBUGGER. PDP-11 DEBUGGERS, WHICH USUALLY HAVE TO BE COGNIZANT OF BYTES, ARE NOTICABLY COMPLICATED BY IT.) BUT -- IS IT COMMON TO NEED TO MODIFY ONLY ONE HALF OF A PRU WORD? OR ONE BIT OF A WORD? COULD SUCH OCCASIONS BE MET BY INLAYING A SMALL PIECE OF CODE WHICH ITSELF DOES THE BYTE OR BIT MODIFICATION?

ONE PARTICULAR IMPLEMENTATION OF THE CHUNK FORMAT IS AS FOLLOWS. THE FIRST WORD (AFTER HTP, ETC. HEADERS) SELECTS EXAMINE OR DEOSIT FUNCTION. COLLINS MAY ASSIGN ANY VALUES CONVENIENT FOR THEM. USE OF A FULL WORD ALLOWS FOR FUTURE EXPANSION OF DEBUGGING FUNCTIONS. NEXT THERE FOLLOW SOME NUMBER N, PERHAPS BETWEEN 4 AND 8, OF ADDRESS AND COUNT FIELDS. IF EACH WORD CONTAINS ONE SUCH PAIR, 12 BITS CAN BE USED FOR A (WORD) ADDRESS AND THE REMAINING 4 BITS FOR A COUNT. WHETHER A COUNT VALUE OF 0 SHOULD MEAN NO WORDS (I.E., NO ACTION ON THIS CHUNK) OR ONE WORD (THUS PERMITTING UP TO 16 WORDS PER CHUNK DATA BLOCK INSTEAD OF ONLY 15) WILL DEPEND ON COLLINS' PREFERENCES. AFTER THESE N ADDRESS AND COUNT FIELDS, THERE FOLLOW N DATA BLOCKS. EACH DATA BLOCK IN SEQUENCE IS ASSOCIATED WITH THE CORRESPONDING ADDRESS AND COUNT FIELDS ABOVE. WHETHER DATA BLOCKS ARE PADDED OUT TO THE MAXIMUM SIZE (15 OR 16 WORDS IN OUR EXAMPLE), OR ARE COMPACTED, IS LEFT TO COLLINS. ALSO, DATA BLOCKS CORRESPONDING TO COUNT FIELDS WHOSE VALUE IS 0 CAN BE OMITTED, IF COLLINS DESIRES.

ONE VARIATION ON THE ABOVE THEME IS TO HAVE ONE FULL WORD FOR ADDRESS AND ONE FULL WORD FOR COUNT. THIS ALLOWS ADDRESSING OF SUB-WORD UNITS, AND/OR ADDRESSING OF PRU CORE ABOVE 4096 WORDS. ALSO, IT PERMITS LARGER DATA BLOCKS THAN 16 WORDS. THE SEPARATE WORDS FOR ADDRESS AND COUNT MAY BE EASIER FOR THE PRU DEBUG CODE TO PARSE. TAKING ADVANTAGE OF SUB-WORD ADDRESSING MAY OVERLY COMPLICATE OR ENLARGE THE PRU DEBUG CODE. THE FINAL DECISION SHOULD BE LEFT TO COLLINS.

ANOTHER VARIATION IS TO INTERMIX THE ADDRESS AND COUNT FIELDS WITH THE DATA BLOCKS, SO THAT EACH CHUNK OCCUPIES AN UNINTERRUPTED PORTION OF THE PACKET.

SOME FUNCTIONS MAY BE PERFORMED BEST BY A SINGLE COMMAND WHICH EXAMINES THE OLD CONTENTS OF A LOCATION AND THEN DEPOSITS NEW DATA. ONE EXAMPLE OF THIS IS COLLECTION OF STATISTICS, FOR WHICH THE VALUE OF A COUNTER MAY HAVE TO BE BOTH EXAMINED AND RESET WITHOUT ANY OTHER PROCESSING INTERVENING.

ANOTHER CONSIDERATION IS THE OPERATION OF THE DEBUG CODE IN THE PRU. CONSIDER THE EXAMPLE DESCRIBED EARLIER, WHERE TWO WORDS IN HTP CODE ARE BEING OVERLAYED. A MULTIPLE DEPOSIT COMMAND IS OF NO HELP IF THE DEBUG CODE MAY BE INTERRUPTED FOR HTP CODE TO RUN. THEREFORE, IT IS HIGHLY DESIRABLE TO ENSURE THAT ONCE THE DEBUG CODE BEGINS A DEPOSIT (OR EXAMINE) OPERATION, IT RUNS TO COMPLETION OF THAT OPERATION, FINISHING ALL CHUNKS OF THAT PACKET, BEFORE ANY OTHER CODE RUNS.

AN ISSUE NEGLECTED IN THE ABOVE DISCUSSION IS THE DIFFERENCE BETWEEN COMMANDS FROM THE STATION TO THE PRU, AND RESPONSES FROM THE PRU TO THE STATION. UPON EXECUTION OF A DEPOSIT COMMAND, THE PRU SHOULD REPLY SOMETHING MEANING THAT THE DEPOSIT HAS BEEN EXECUTED. SINCE HEADERS NOT

DISCUSSED HEREIN PROVIDE SEQUENCE NUMBERS; THIS REPLY NEED BE NOTHING MORE THAN AN ACKNOWLEDGEMENT. IF OTHER LEVELS OF PRU CODE GENERATE THE ACKNOWLEDGEMENT BEFORE (OR IN PARALLEL WITH) DELIVERY TO THE DEBUG CODE; THEN A SPECIFIC "OK" REPLY MUST BE SENT. SINCE CONNECTIONS FOR DEBUGGING WILL BE SEQUENCED (MESSAGES WILL NOT GET OUT OF ORDER; ONLY ONE OUTSTANDING UNACKNOWLEDGED MESSAGE MAY EXIST AT ANY MOMENT); NO FURTHER SEQUENCE NUMBER OR ADDRESSES OR COUNTS NEED APPEAR IN THE "OK."

IT IS NOT IMMEDIATELY CLEAR WHETHER THE REPLY SHOULD BE GENERATED AND TRANSMITTED BEFORE OR AFTER THE DEPOSIT IS PERFORMED. DISCUSSION OF THIS ISSUE IS NEEDED. IN X-NET, FOR INSTANCE, THE PROTOCOL EXPLICITLY STATES THAT RESPONSES FROM THE PDP-11 TO THE PDP-10 OF "RESUMED RUNNING OF PROCESS," "SINGLE STEPPED PROCESS," AND "PROCEEDED BREAKPOINT" BE SENT BEFORE ACTUALLY DOING SO. THIS IS BECAUSE THE PROCESS MAY IMMEDIATLY HALT, HIT A BREAKPOINT, OR STOP IN SOME OTHER MANNER. IF HTP CODE IN THE PRU IS BEING OVERLAID, FOR INSTANCE, IT MAY BE OF CONCERN WHETHER THE "DEPOSIT PERFORMED" REPLY IS TRANSMITTED WITH THE PREVIOUS OR THE NEW CONTENTS. ONE INTERESTING ALTERNATIVE IS TO ASSIGN A THIRD VALUE TO THE EXAMINE/DEPOSIT VARIABLE, DISTINGUISHING BETWEEN DEPOSIT FIRST AND REPLY FIRST.

UPON EXECUTION OF AN EXAMINE COMMAND, THE PRU SHOULD PACK THE MEMORY CONTENTS INTO THE APPROPRIATE DATA BLOCKS IN THE PACKET, AND SEND THE PACKET TO THE STATION. REPEATING THE ADDRESS AND COUNT FIELDS IN THE EXAMINE COMMAND IS NOT NECESSARY, SINCE THE STATION KNOWS THE VALUES OF THESE FIELDS IN THE EXAMINE COMMAND WHICH ELICITED THE RESPONSE. IF IMPLEMENTATION IS EQUALLY EASY EITHER WAY, THOUGH, REPEATING THESE FIELDS GIVES ONE MORE ITEM WHICH CAN BE CHECKED FOR CONSISTENCY. ANOTHER CONSIDERATION IS THAT NOT REPEATING THESE FIELDS WOULD ALLOW MORE MEMORY TO BE EXAMINED PER PACKET.

IN X-NET, STATE INFORMATION IS ALSO TRANSFERRED BETWEEN THE PDP-11 AND THE PDP-10. ADVICE FROM COLLINS IS NEEDED ON WHETHER SIMILAR INFORMATION IN THE PRU CAN BE OBTAINED FROM MEMORY LOCATIONS. STATUS REGISTERS, ACCUMULATORS, ETC. CAN BE READ BY AN INLAY AND TEMPORARILY STORED IN ANY CONVENIENT MEMORY LOCATION, AND THEN EXAMINED BY A NORMAL EXAMINE COMMAND. BUT IS THIS MECHANISM ADEQUATE? ARE SOME STATUS CONDITIONS MORE EASILY ACCESSED EITHER BY FUNCTIONS IN ADDITION TO EXAMINE AND DEPOSIT, OR BY "PSEUDO-ADDRESSES" BEYOND THE HIGHEST ACTUAL CORE LOCATION? THE EXPERIENCE OF COLLINS PROGRAMMERS IS OF UTMOST IMPORTANCE IN THIS DESIGN ISSUE.

II IMPLEMENTATION

COLLINS PLANS TO IMPLEMENT CROSS-RADIO DEBUGGING OF PRUS AS EXECUTABLE PACKETS. A PACKET OF DEBUGGING INFORMATION WILL BE SPECIALLY MARKED AS CODE FOR EXECUTION. (THIS MAY ALSO BE USED FOR OTHER FUNCTIONS, SUCH AS STATISTICS COLLECTION.) UPON RECEIPT BY THE PRU, CONTROL WILL BE TRANSFERRED TO THE RECEIVED PACKET AND ITS CONTENTS EXECUTED AS INSTRUCTIONS LOCATED IN THE PACKET BUFFER. THESE INSTRUCTIONS MUST EITHER BE POSITION INDEPENDENT CODE OR REFERENCE BASE PAGE. THE CODE WILL HAVE ACCESS TO ALL MEMORY LOCATIONS AND REGISTERS AND MAY EXECUTE

Digitized by the Internet Archive
in 2025 with funding from
Amateur Radio Digital Communications, Grant 151

https://archive.org/details/temporary-note-145

ANY IMPLEMENTED IMP-16 INSTRUCTIONS. THIS INCLUDES MASKING (DISABLING) INTERRUPTS; CALLING OPERATING SYSTEM ROUTINES TO SEND PACKETS; AND PREPARING ANY SORT OF PACKETS IT MAY NEED TO SEND. THE MAJOR LIMITATION IS THAT THE CODE MUST FIT IN A PACKET; WHICH IS 127 WORDS INCLUDING THE HEADER.

LET US ASSUME THAT THE PACKET SIZE CONSTRAINT ON CODE IS NOT SERIOUS. EACH OF THE FUNCTIONS DISCUSSED IN SECTION I ABOVE CAN BE PROVIDED: EXAMINE AND/OR DEPOSIT SCATTERED WORDS AND/OR BLOCKS OF WORDS; EXAMINE/DEPOSIT OF PORTIONS OF WORDS; DISABLING INTERRUPTS SO ALL DEBUGGING OPERATIONS IN THE PACKET ARE PERFORMED WITHOUT INTERRUPTION. ALSO; A REPLY CAN BE SENT (AND ACKNOWLEDGEMENT AWAITED; WE BELIEVE) BEFORE A NEW VALUE IS DEPOSITED IN MEMORY.

THEREFORE THE CRUCIAL QUESTION IS; HOW MUCH CODE IS NECESSARY TO PERFORM THE DESIRED FUNCTIONS? THIS IS AN ISSUE WHICH WE RELY UPON COLLINS TO RESOLVE. ALSO; FOR OTHER APPLICATIONS OF REMOTE DEBUGGING OF COMPUTERS; WE POINT THIS OUT AS A CENTRAL QUESTION. GIVEN THAT THE EXECUTABLE DEBUGGING PACKET HAS FULL ACCESS TO ALL MEMORY LOCATIONS AND REGISTERS; AND THAT IT CAN DISABLE INTERRUPTS -- HOW SMALL CAN THE CODE BE MADE?